



IPW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Gluch
Serial No.: 10/788,602
Filed: February 27, 2004
For: Method for Communication with a Test System for Integrated Circuits

Docket No.: 2002 P 50357 US
Art Unit: 2862
Examiner: TBD

Mail Stop: Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF CERTIFIED COPY OF PRIORITY DOCUMENT

Dear Sir:

Attached please find a certified copy of the foreign application from which priority is claimed for this case:

Country: Germany
Application Number: 10309208.0
Filing Date: February 28, 2003

Respectfully submitted,

Ira S. Matsil
Attorney for Applicant
Reg. No. 35,272

June 16, 2004

Slater & Matsil, L.L.P.
17950 Preston Road, Suite 1000
Dallas, Texas 75252
(972) 732-1001 - Tel
(972) 732-9218 - Fax

BUNDESREPUBLIK DEUTSCHLAND



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 103 09 208.0

Anmeldetag: 28. Februar 2003

Anmelder/Inhaber: Infineon Technologies AG, 81669 München/DE

Bezeichnung: Verfahren zur Kommunikation mit einem Testsystem
für integrierte Schaltungen

IPC: G 01 R, H 01 L

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 18. Februar 2004
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Remus



5 **Verfahren zur Kommunikation mit einem Testsystem für
integrierte Schaltungen**

Die Erfindung betrifft ein Verfahren zur Kommunikation mit
einem Testsystem für integrierte Schaltungen, bei dem Befehle
10 eines High-Level-Programms abgearbeitet werden und dabei ein
Low-Level-Programm Testsignale erzeugt, die an die zu testende
integrierte Schaltung übermittelt werden, und bei dem
Reaktionssignale von der integrierten Schaltung durch das Low-
Level-Programm als Reaktionen an das High-Level-Programm
15 übermittelt werden.

Ein Verfahren der eingangs genannten Art ist beispielsweise aus
der US 6,304,095 B1 bekannt. Darin wird eine Halbleiter-
messvorrichtung mit der Fähigkeit zur dynamischen Änderung von
Prüfkriterien beschrieben.

20 Darin wird auch der allgemeine Aufbau eines Testsystems und
programmseitige Ablauf beschrieben. Ein Tester stellt eine
Hardware dar, die geeignet ist, für die zu testende integrierte
Schaltung geeignete Testsignale zu generieren und von der zu
testenden integrierten Schaltung Reaktionssignale zu empfangen
25 und zur Auswertung weiterzuleiten oder zu speichern, um damit
eine Fehlerauswertung zu ermöglichen.

Der Tester wird dabei von einer Software in einem
maschinennahen Code, nachfolgend als Low-Level-Programm,
bezeichnet, gesteuert. Dieses Low-Level-Programm wird von einem
30 Kernprogramm gesteuert, das mit verschiedenen test-,
schaltkreis- und nutzerspezifische Programmteilen kommuniziert,
beispielsweise mit einem Testplan, einem User-Interface oder
einem Messalgorithmus. Die Kommunikation besteht dabei in der
Generierung von Testbefehlen, die in dem Low-Level-Programm in

Signale gewandelt werden und in einem Empfang von Reaktionen, die von dem Low-Level-Programm aus Reaktionssignalen erzeugt werden.

Das Kernprogramm mit diesen Programmteilen besteht aus Befehlen
5 einen höheren Programmiersprache und wird nachfolgend als High-Level-Programm bezeichnet.

Da das High-Level-Programm und dabei zumindest das Kernprogramm mit dem Low-Level-Programm kommunizieren muss und das Low-Level-Programm von der Hardware des Testers abhängt, also
10 gerätespezifisch ist, hat zwangsläufig jedes Testsystem, sein eigenes spezifisches High-Level-Programm. Dies bedingt, dass sich das High-Level-Programm von Herstellerfirma zu Herstellerfirma, oft aber auch noch von Gerätegeneration zu Gerätegeneration unterscheidet. Der Nachteil besteht dabei
15 darin, dass der Programmierer des Testsystems das High-Level-Programm bei jedem Gerät oder jeder Generation neu erlernen muss. Auch entspricht das High-Level-Programm oft nicht den Anforderungen für die konkrete Testaufgabe.

Die Aufgabe der Erfindung besteht somit darin, den Zeitaufwand
20 für die Herstellung der Bedienbereitschaft von Testsystemen dadurch zu senken, dass ein im wesentlichen bei verschiedenen Testsystemen einheitliches High-Level-Programm eingesetzt wird.

Gemäß der Erfindung wird die Aufgabe dadurch gelöst, dass ein selbständiges, für das Testsystem spezifisches Interface-
25 Programm derart bereitgestellt wird, dass ein High-Level-Code in einen oder mehrere, der Funktionalität des High-Level-Codes entsprechende Low-Level-Codes und ein oder mehrere Low-Level-Codes in einen oder mehrere High-Level-Codes, die der Funktionalität des Low-Level-Codes entsprechen, umgesetzt wird
30 und dass das High-Level-Programm mit dem Low-Level-Programm ausschließlich über das Interface-Programm verbunden ist und umgekehrt.

Dabei können die High-Level- wie auch die Low-Level-Codes

sowohl Programm- als auch Datencodes darstellen.

Mit diesem Verfahren wird somit das High-Level-Programm von dem Low-Level-Programm unabhängig gestaltet. Es ist lediglich erforderlich, für Testsysteme verschiedener Hersteller oder für
5 verschiedene Gerätegenerationen jeweils spezifische Interface-Programme bereit zu stellen. Die High-Level-Codes, die das Interface-Programm "verstehen", können dabei unabhängig von dem jeweiligen Testsystem gestaltet werden.

In einer zweckmäßigen Ausgestaltung der Erfindung ist
10 vorgesehen, dass das High-Level-Programm einen von dem Testsystem unabhängigen Vorrat an High-Level-Codes aufweist. Damit wird es möglich, ein High-Level-Programm bereit zu stellen, dass unabhängig von dem eingesetzten Testsystem eine einheitliche Benutzeroberfläche aufweist. Jede Bedienperson
15 kennt dann nach einem einmaligen Lernvorgang alle oder die wesentlichsten High-Level-Codes und kann damit geringem Zeitaufwand und geringer Fehlerwahrscheinlichkeit die Bedienbereitschaft von Testsystemen herstellen.

Die Erfindung soll nachfolgend anhand eines Ausführungs-
20 beispieles näher erläutert werden. In den zugehörigen Zeichnungen zeigt

Fig. 1 eine schematische Darstellung der Kommunikation mit einem Testsystem für integrierte Schaltungen nach dem Stand der Technik und

25 Fig. 2 eine schematische Darstellung der erfindungsgemäßen Kommunikation mit einem Testsystem für integrierte Schaltungen.

Wie in Fig. 1 dargestellt, ist in einem ersten Testsystem 1 ein
erstes Low-Level-Programm 2 implementiert. Das erste Testsystem
30 1 wird von dem ersten Low-Level-Programm 2 gesteuert. Das erste Low-Level-Programm 2 seinerseits kommuniziert mit einem ersten High-Level-Programm 3. Dabei wird ein High-Level-Code des

ersten High-Level-Programms 3 durch das erste High-Level-
Programm 3 in einen oder mehrere, der Funktionalität des High-
Level-Codes entsprechende Low-Level-Codes des ersten Low-Level-
Programms 2 umgesetzt. Umgekehrt werden in dem ersten High-
5 Level-Programm 3 ein oder mehrere Low-Level-Codes des ersten
Low-Level-Programms 2 in einen oder mehrere High-Level-Codes
des ersten High-Level-Programms 3, die der Funktionalität des
Low-Level-Codes entsprechen, umgesetzt.

In gleicher Weise ist einem zweiten Testsystem 4 ein zweites
10 Low-Level-Programm 5 implementiert. Das zweite Testsystem 4
wird von dem zweiten Low-Level-Programm 5 gesteuert. Das zweite
Low-Level-Programm 5 seinerseits kommuniziert mit einem zweiten
High-Level-Programm 6. Dabei wird ein High-Level-Code des
zweiten High-Level-Programms 6 durch das zweite High-Level-
15 Programm 6 in einen oder mehrere, der Funktionalität des High-
Level-Codes entsprechende Low-Level-Codes des zweiten Low-
Level-Programms 5 umgesetzt. Umgekehrt werden in dem zweiten
High-Level-Programm 6 ein oder mehrere Low-Level-Codes des
zweiten Low-Level-Programms 5 in einen oder mehrere High-Level-
20 Codes des zweiten High-Level-Programms 6, die der
Funktionalität des Low-Level-Codes entsprechen, umgesetzt.

Wie daraus ersichtlich wird, ist somit spezifisch für das erste
Testsystem 1 ein erstes High-Level-Programm 3 und für das
zweite Testsystem 4 ein zweites High-Level-Programm 6
25 erforderlich. Bei der Herstellung der Bedienungsbereitschaft
des einen und des anderen Testsystems muss die Bedienperson
zwischen mindestens zwei verschiedenen Software-Tools wechseln,
die vorher zu lernen sind. Auch ist der Support mit geeigneten
Setups für die beiden oder weitere Testsysteme aufwändig. Oft
30 ist auch das High-Level-Programm nicht vollständig an die
Testerfordernisse angepasst. Anpassungen und Änderungen sind
aber oft nur sehr begrenzt möglich.

Die in Fig. 2 dargestellte erfindungsgemäße Lösung stellt nun für das erste Testsystem 1 spezifisches erstes Interface-Programm 7 derart bereit, dass ein High-Level-Code des einheitlichen High-Level-Programms 8 in einen oder mehrere, der Funktionalität des High-Level-Codes entsprechende Low-Level-Codes des ersten Low-Level-Programms 2 und ein oder mehrere Low-Level-Codes des ersten Low-Level-Programms 2 in einen oder mehrere High-Level-Codes des einheitlichen High-Level-Programms 8, die der Funktionalität des Low-Level-Codes entsprechen, umgesetzt wird.

In gleicher Weise wird für das zweite Testsystem 4 ein spezifisches zweites Interface-Programm 9 derart bereitgestellt, dass ein High-Level-Code des einheitlichen High-Level-Programms 8 in einen oder mehrere, der Funktionalität des High-Level-Codes entsprechende Low-Level-Codes des zweiten Low-Level-Programms 5 und ein oder mehrere Low-Level-Codes des zweiten Low-Level-Programms 5 in einen oder mehrere High-Level-Codes des einheitlichen High-Level-Programms 8, die der Funktionalität des Low-Level-Codes entsprechen, umgesetzt wird.

Grundsätzlich ist das einheitliche High-Level-Programm 8 mit dem ersten Low-Level-Programm 2 ausschließlich über das erste Interface-Programm 7 verbunden ist und umgekehrt. In gleicher Weise ist das einheitliche High-Level-Programm 8 mit dem zweiten Low-Level-Programm 5 ausschließlich über das zweite Interface-Programm 9 verbunden ist und umgekehrt.

Dabei können die High-Level- wie auch die Low-Level-Codes sowohl Programm- als auch Datencodes darstellen.

Mit diesem Verfahren wird somit das einheitliche High-Level-Programm 8 von dem jeweiligen Low-Level-Programm 2, 5 unabhängig gestaltet. Es ist lediglich erforderlich, für Testsysteme verschiedener Hersteller oder für verschiedene Gerätegenerationen jeweils spezifische Interface-Programme 7, 9

bereit zu stellen. Die High-Level-Codes, die das Interface-Programm "versteht", können dabei unabhängig von dem jeweiligen Testsystem gestaltet werden.

5 Der Vorteil der Erfindung ist, dass man sich die Funktionalität der Testsysteme 1, 4 zunutze machen kann, und eine Schnittstelle auf der "Seite" des jeweiligen Interface-Programms 7, 9, die dem einheitlichen High-Level-Programm 8 "zugewandt" ist, mit gleichen Funktionen auf allen Testsystemen 1, 4 zur Verfügung stellt. Aufbauend auf diesen Funktionen kann
10 nun ein einheitliches High-Level-Programm 8 bereit gestellt werden, das für alle Testsysteme 1, 4 gleich ist. Somit können vormals unterschiedliche Befehle für ein und dieselbe Funktion, z.B. "power supply = 5V" oder "set power supply 5V" durch ein und den selben Befehl, z.B. "set power supply 5V" ausgedrückt
15 werden.

5 **Verfahren zur Kommunikation mit einem Testsystem für
integrierte Schaltungen**

Bezugzeichenliste

- | | | |
|----|---|-----------------------------------|
| 10 | 1 | erstes Testsystem |
| | 2 | erstes Low-Level-Programm |
| | 3 | erstes High-Level-Programm |
| | 4 | zweites Testsystem |
| | 5 | zweites Low-Level-Programm |
| 15 | 6 | zweites High-Level-Programm |
| | 7 | erstes Interface-Programm |
| | 8 | einheitliches High-Level-Programm |
| | 9 | zweites Interface-Programm |

M

5 **Verfahren zur Kommunikation mit einem Testsystem für
integrierte Schaltungen**

Patentansprüche

10 1. Verfahren zur Kommunikation mit einem Testsystem für
integrierte Schaltungen, bei dem Befehle eines High-Level-
Programms abgearbeitet werden und dabei ein Low-Level-Programm
Testsignale erzeugt, die an die zu testende integrierte
15 Schaltung übermittelt werden, und bei dem Reaktionssignale von
der integrierten Schaltung durch das Low-Level-Programm als
Reaktionen an das High-Level-Programm übermittelt werden,
dadurch gekennzeichnet, dass ein selbständiges,
für das Testsystem (1; 4) spezifisches Interface-Programm (7;
20 9) derart bereitgestellt wird, dass ein High-Level-Code in
einen oder mehrere, der Funktionalität des High-Level-Codes
entsprechende Low-Level-Codes und ein oder mehrere Low-Level-
Codes in einen oder mehrere High-Level-Codes, die der
Funktionalität des Low-Level-Codes entsprechen, umgesetzt wird
und dass das High-Level-Programm (8) mit dem Low-Level-Programm
25 (2; 5) ausschließlich über das Interface-Programm (7; 9)
verbunden ist und umgekehrt.

2. Verfahren nach Anspruch 1, dadurch gekenn-
zeichnet, dass das High-Level-Programm (8) einen von dem
30 Testsystem (1; 4) unabhängigen Vorrat an High-Level-Codes
aufweist..

5 **Verfahren zur Kommunikation mit einem Testsystem für
integrierte Schaltungen**

Zusammenfassung

- 10 Der Erfindung, die ein Verfahren zur Kommunikation mit einem
Testsystem für integrierte Schaltungen betrifft, bei dem
Befehle eines High-Level-Programms abgearbeitet werden und
dabei ein Low-Level-Programm Testsignale erzeugt, die an die zu
testende integrierte Schaltung übermittelt werden, und bei dem
15 Reaktionssignale von der integrierten Schaltung durch das Low-
Level-Programm als Reaktionen an das High-Level-Programm
übermittelt werden, liegt die Aufgabe zugrunde, den Zeitaufwand
für die Herstellung der Bedienbereitschaft von Testsystemen
dadurch zu senken, dass ein im wesentlichen bei verschiedenen
20 Testsystemen einheitliche High-Level-Programm eingesetzt wird.
Dies wird dadurch gelöst, dass ein selbständiges, für das
Testsystem spezifisches Interface-Programm derart
bereitgestellt wird, dass ein High-Level-Code in einen oder
mehrere, der Funktionalität des High-Level-Codes entsprechende
25 Low-Level-Codes und ein oder mehrere Low-Level-Codes in einen
oder mehrere High-Level-Codes, die der Funktionalität des Low-
Level-Codes entsprechen, umgesetzt wird und dass das High-
Level-Programm mit dem Low-Level-Programm ausschließlich über
das Interface-Programm verbunden ist und umgekehrt. (Fig. 2)

Fig. 1 (Stand der Technik)

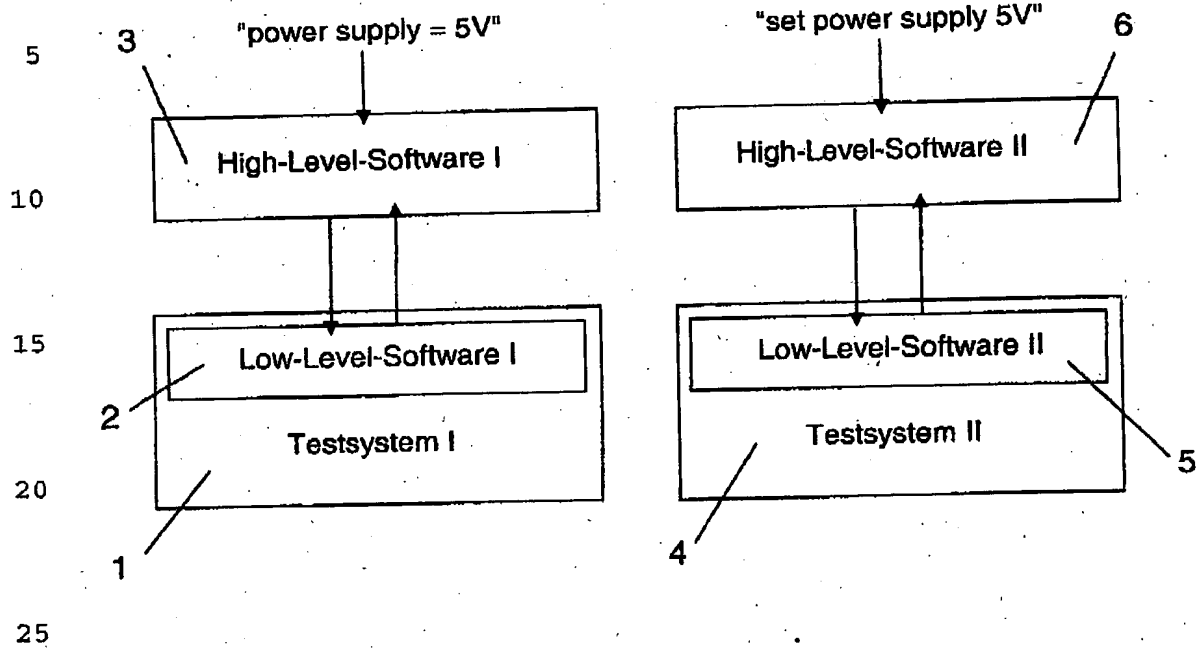


Fig. 2

